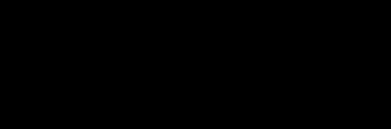


Quartz 2D and PDF

Session 501

Quartz 2D and PDF

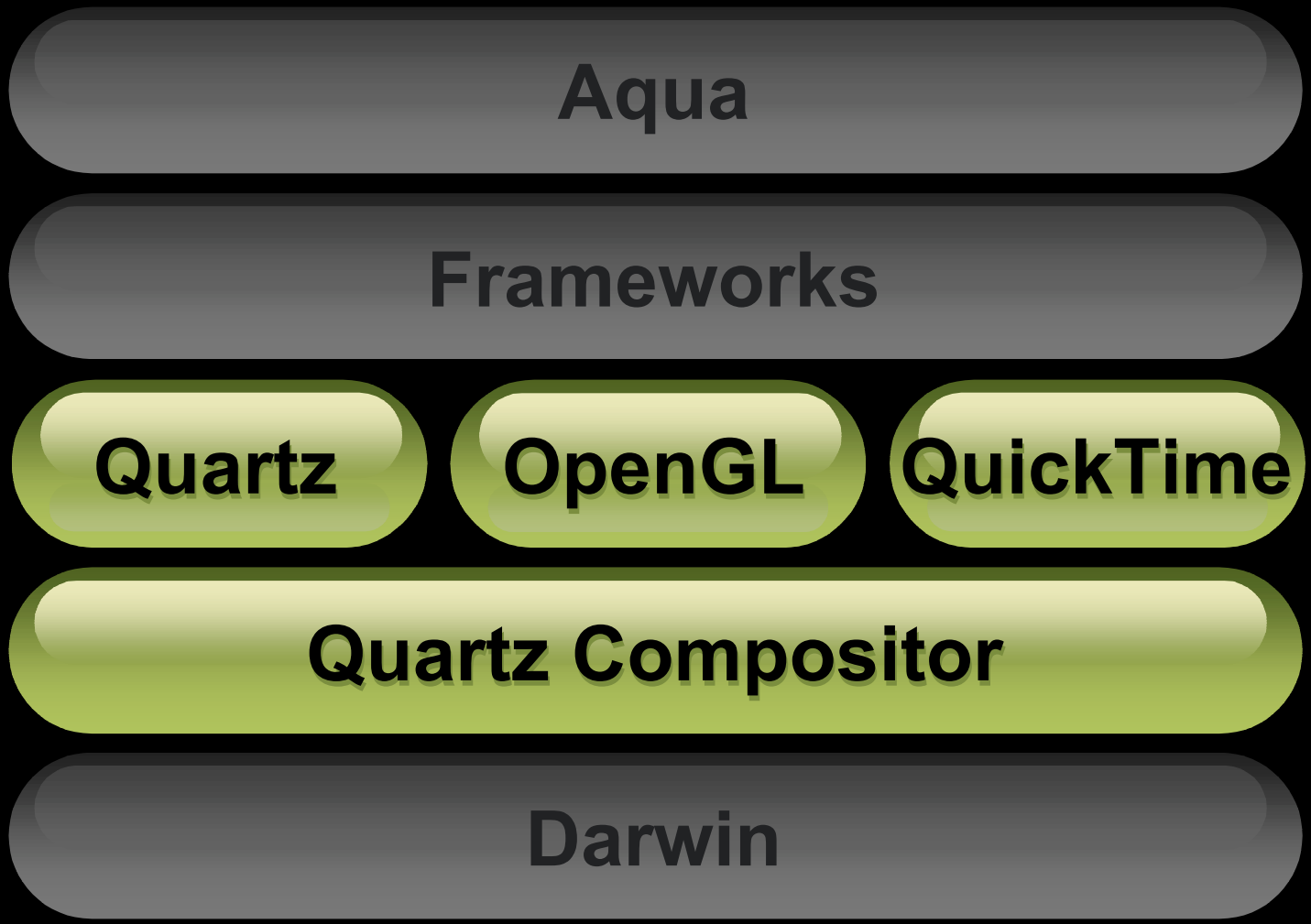
Derek B Clegg
Quartz 2D Engineering

Agenda

- What is Quartz?
- Quartz Architecture
- Quartz API
- Demo

What Is Quartz?

- Quartz is the underlying graphics system on Mac OS X
- Two components
    - Quartz 2D
    - Quartz Compositor


Aqua
Frameworks
Quartz
OpenGL
QuickTime
Quartz Compositor
Darwin

Quartz Compositor

- All window management and compositing
- Learn more in session 503, “Exploring the Quartz Compositor”

Quartz 2D

- Low-level, lightweight 2D rendering library
- Resolution-independent
- Device-independent
- Uses ATS for font management
- Uses ColorSync for color management

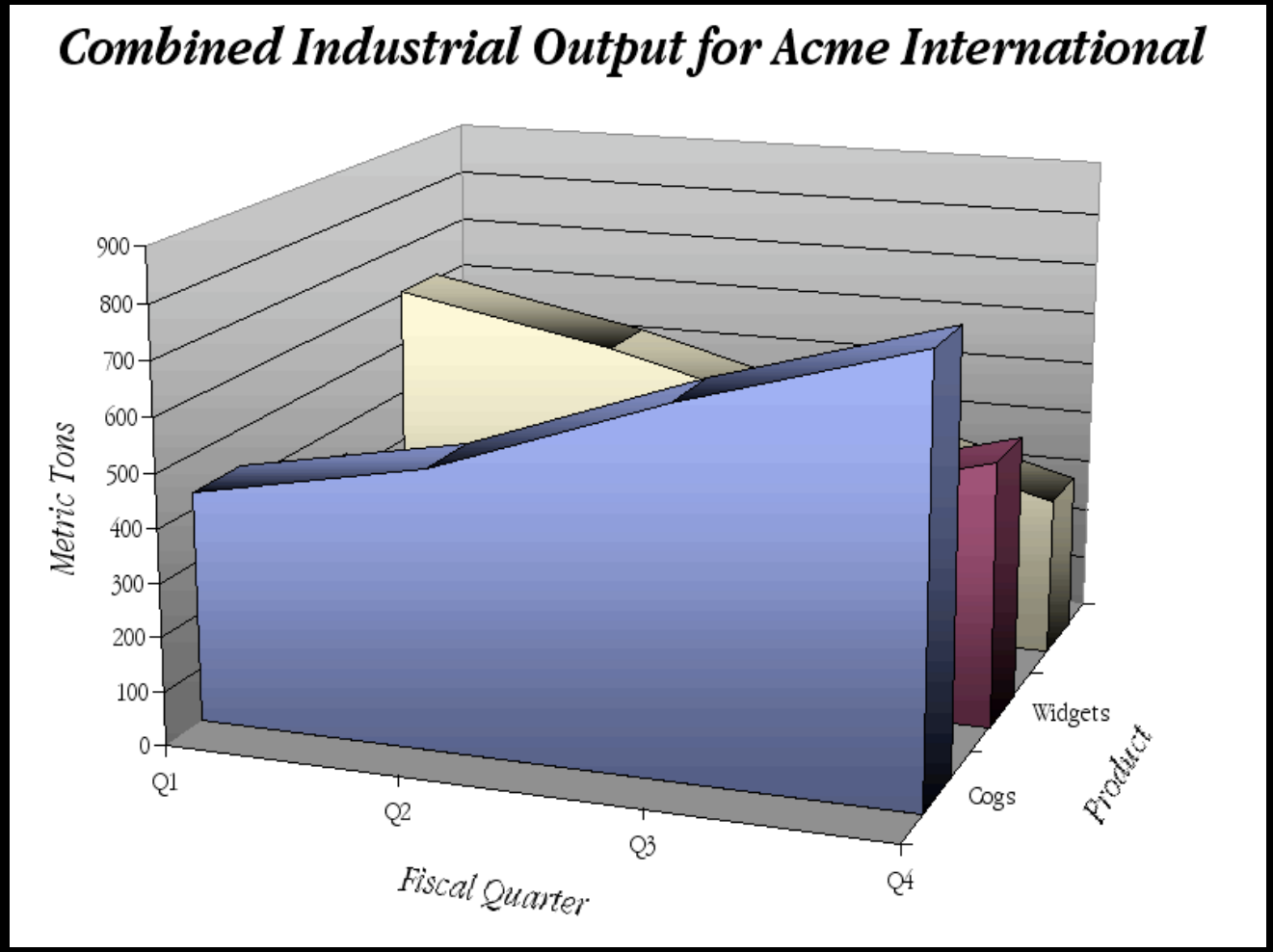
Who Uses Quartz 2D?

- High-level application services
  - Cocoa
  - Carbon
  - Java
- More and more third-party applications
- Your application via the CoreGraphics framework

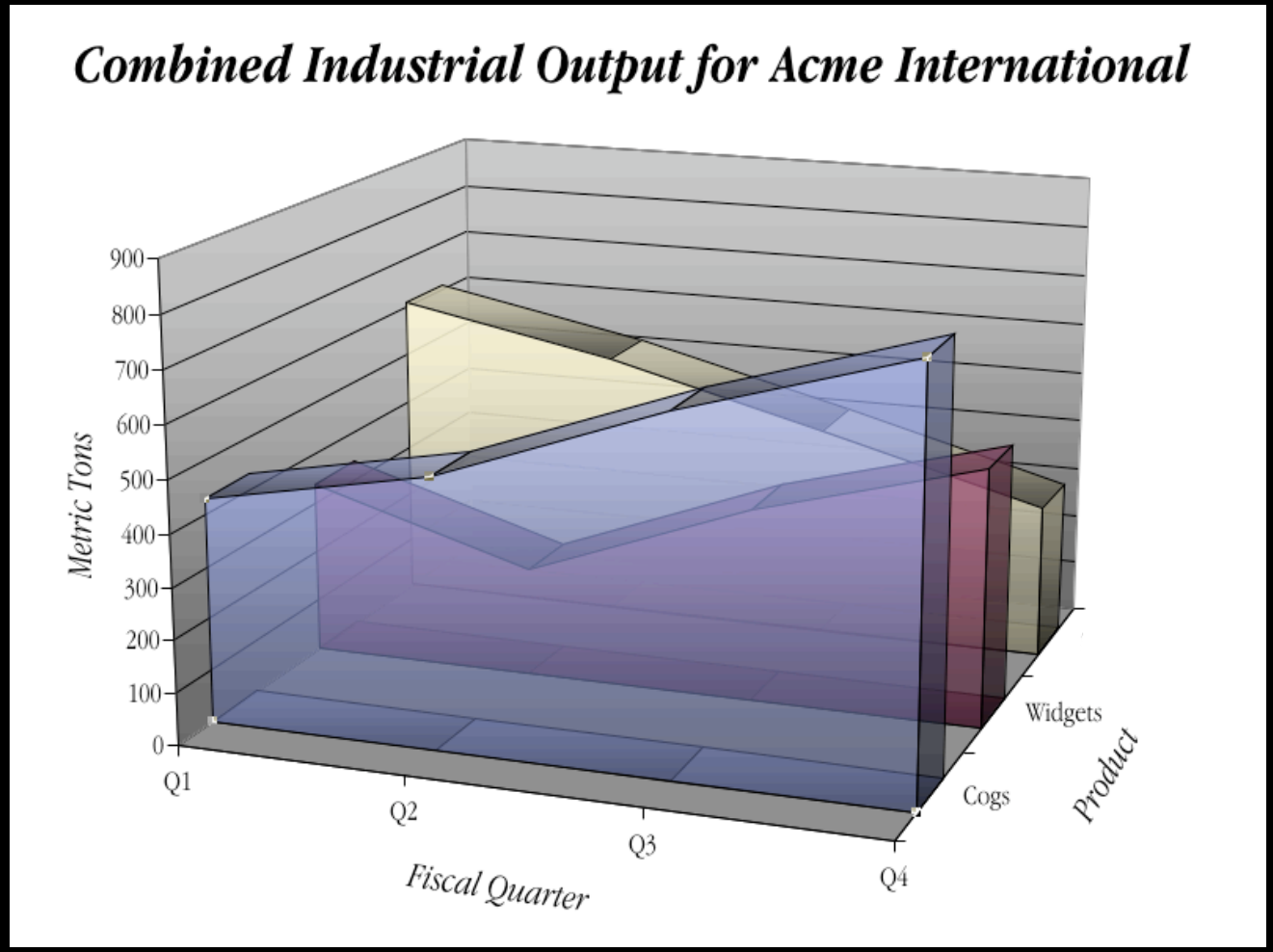
Advantages of Quartz 2D

- High-quality 2D graphics
  - Anti-aliased rendering
  - Transparency
- Offscreen rendering
- PDF document import
- PDF document export

Classical QuickDraw



Quartz 2D


Combined Industrial Output for Acme International
Metric Tons
900
800
700
600
500
400
300
200
100
0
Q1
Q2
Q3
Q4
Fiscal Quarter
Widgets
Cogs
Product

CoreGraphics Types

- Focus on additions since last year
- Some exciting new API
  (if API can be called exciting)

CoreGraphics Types (Cont.)

- Contexts
- Paths
- Fonts
- Images
- PDF Documents

CoreGraphics Types (Cont.)

- Colors/Color Spaces
- Patterns
- Shadings
- Geometry/Affine transforms
- Data Managers

CGContexts

- Device-independent abstraction
- Maintains graphics state information
    - For example, colors, line styles, and so on
    - Graphics state can be saved and restored

CGContexts (Cont.)

- Supported contexts
  - Window—created by Cocoa or Carbon
  - PostScript—created by printing system
  - PDF—**CGPDFContextCreate**
    - PDF file creation
  - Bitmap—**CGBitmapContextCreate**
    - Off-screen rendering

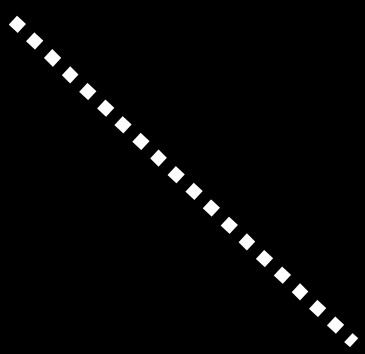
Drawing Primitives

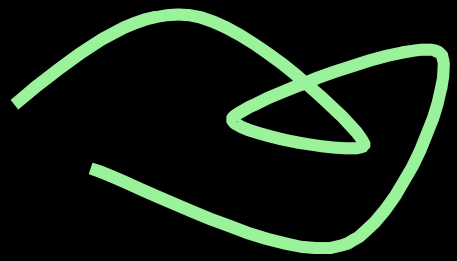
- CGPath: Vector Geometry
- CGFont: Text
- CGImage: Images
- CGPDFDocument: PDF Document Import
- NEW CGPattern: Repeated drawing
- NEW CGShading: Gradients

Demo

Vector Geometry

Vector Geometry





Vector Geometry

- Path Definition
  - 

    CGPathCreateMutable
    CGPathMoveToPoint
    CGPathAddLineToPoint
    - And so on
- Path Conveniences
    CGPathAddArc, etc.
    - DTS sample code

Vector Geometry

- Path Drawing
  - **CGContextAddPath**
  - **CGContextFill**
  - **CGContextStroke**
    - And so on

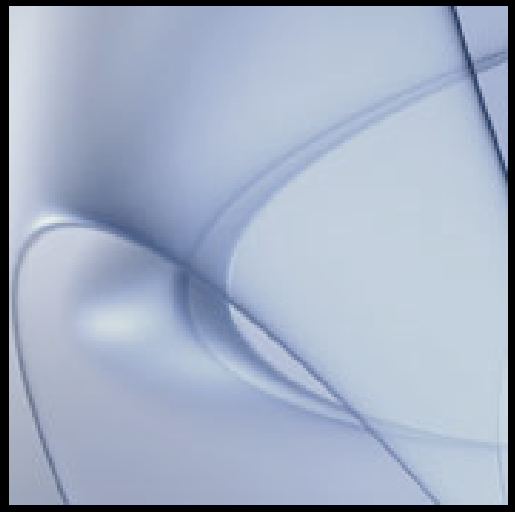
Vector Geometry

- Drawing Operations
  - Fill, stroke, clip
- Line stroking parameters
  - Line width, line join, line cap, miter limit, line dash

Path Examples (Cont.)

Create a
Closed Path

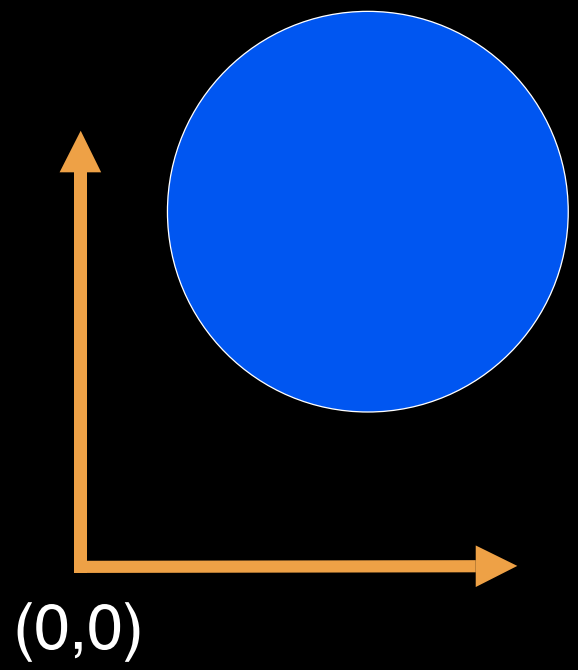
CGContextAddPath
CGContextClip



Continue Drawing
(Bitmap Example)

Resulting
Clip

Working With Paths



path = CGPathCreateMutable

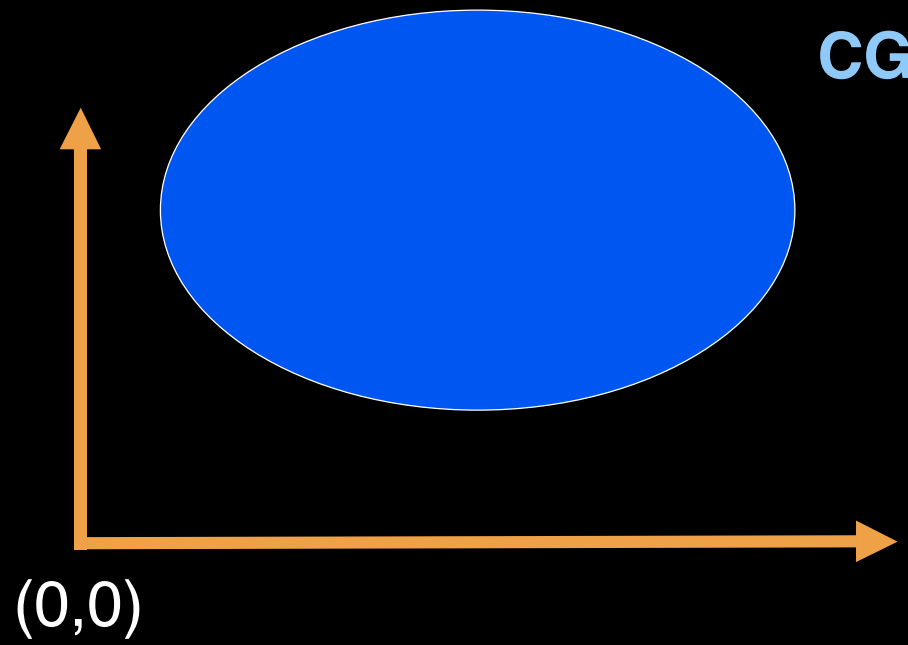
CGPathAddArc

CGPathCloseSubpath

CGContextAddPath(…, path)

CGContextFill

Working With Paths

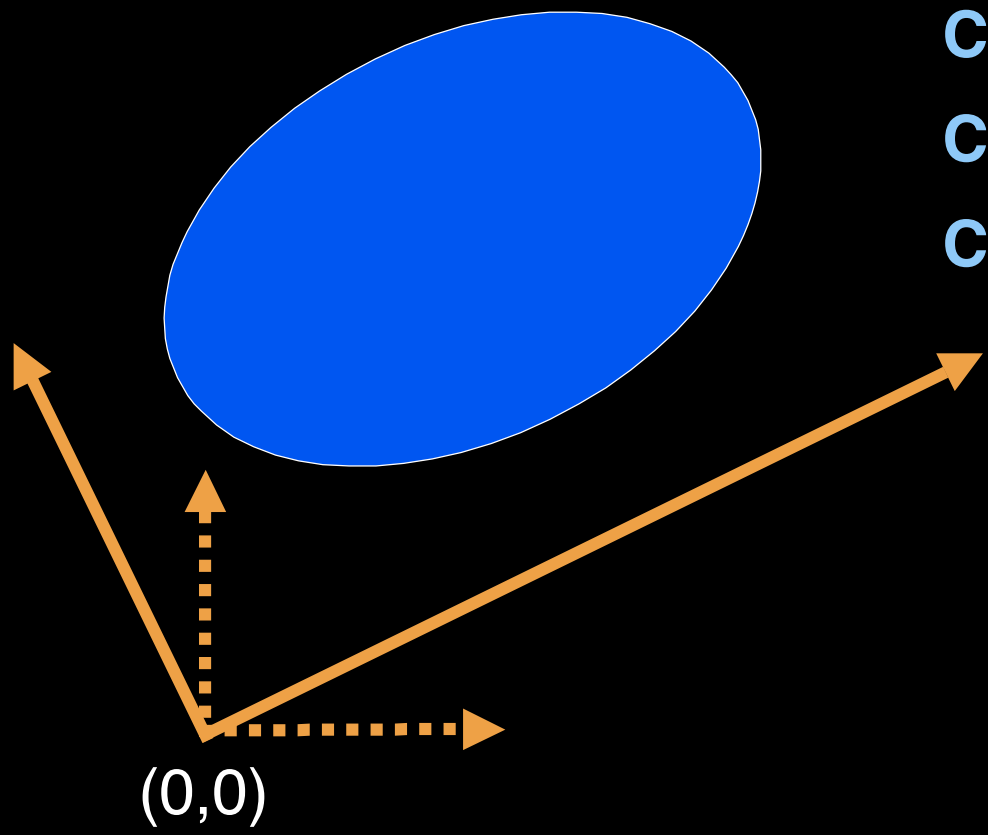


CGContextScaleCTM

CGContextAddPath(…, path)

CGContextFill

Working With Paths



CGContextRotateCTM

CGContextScaleCTM

CGContextAddPath(…, path)

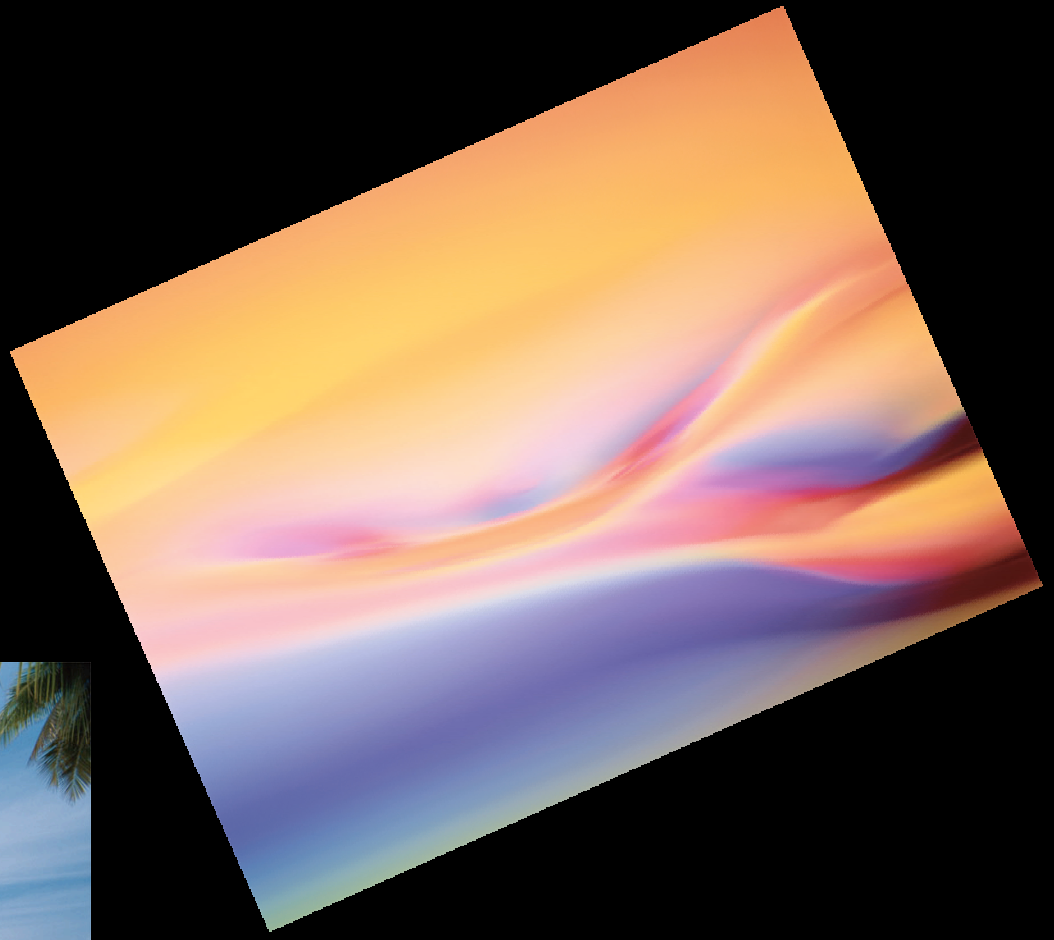
CGContextFill

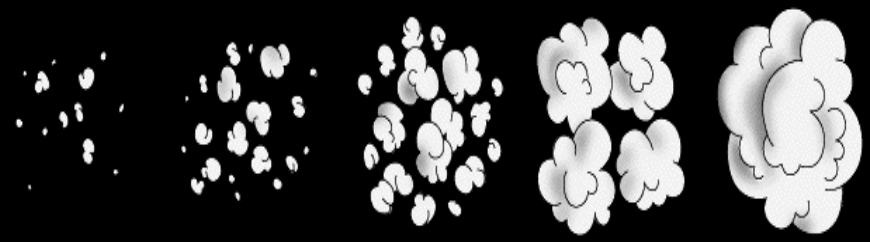

Text
Welcome
안녕하세요
Stroked
サンノゼへようこそ
Rotated

Text

- CGFont represents system fonts
- TrueType, Type1, and CID font support
- Use ATSUI or Cocoa for Unicode and layout support

Images





Images

- CGImage represents image data
- Support for many color spaces
  - RGB, CMYK, ICC profiles and more
- Support for alpha channel
  - Premultiplied or not
- Support for image masks

Images (Cont.)

- Support for extended bit depths
  - 1 bit to 32 bits per component
- Support for high-fidelity color
  - ColorSync matches to destination
  - Image must have calibrated colors

Using Images

- Create a data provider to supply image data
- Create a **CGImage** using **CGImageCreate**
- Draw using **CGContextDrawImage**

Example

```
provider = CGDataProviderCreateWithData(NULL, data, size, NULL);

image = CGImageCreate(w, h, bitsPerSample, bitsPerPixel, bytesPerRow,
    colorspace, alphaInfo, provider, NULL, false,
    kCGRenderingIntentDefault);

CGDataProviderRelease(provider);

rect = CGRectMake(x, y, w, h);

CGContextDrawImage(context, rect, image);

CGImageRelease(image);
```

Custom Data Providers

Arbitrary Processing of Image Data

Example

```
static const CGDataProviderCallbacks callbacks = { &planarGetBytes,
    &planarSkipBytes, &planarRewind, &planarRelease };

provider =  CGDataProviderCreate(info, &callbacks);

image = CGImageCreate(w, h, bitsPerSample, bitsPerPixel, bytesPerRow,
    colorspace, alphaInfo, provider, NULL, false,
    kCGRenderingIntentDefault);

...
```

Custom Data Providers

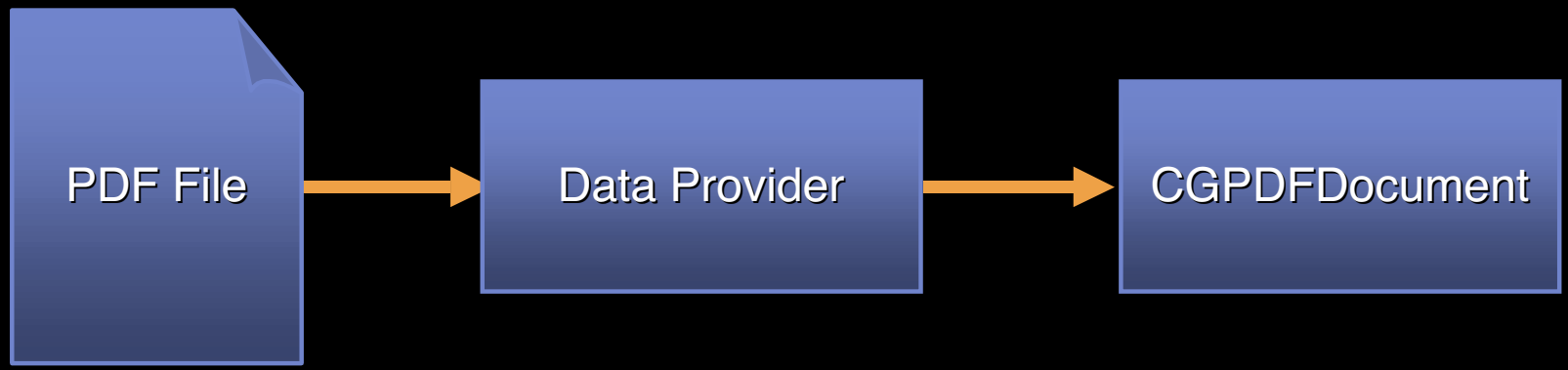
- Other image formats
  - For example, Cineon ™
- Image filters
  - For example, downsampling image data
- Special access
  - For example, QuickTime data provider

Custom Data Providers

- **CGImageCreateWithJPEGDataProvider**
  - Draws image to screen
  - Streams JPEG data to PDF
  - Big savings in file size
- **CGImageCreateWithPNGDataProvider**

PDF Documents

CGPDFDocument Contains PDF Data



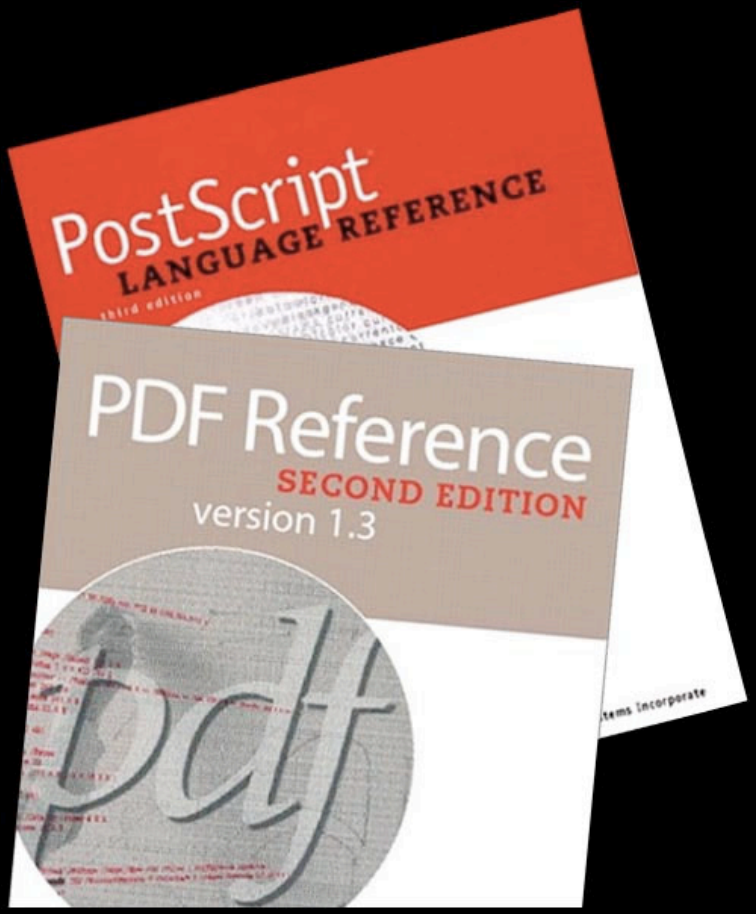
PDF Documents

- Accessors
  - Number of pages
  - Bounding box information
  - And so on
- Draw with **CGContextDrawPDFDocument**

Example

```
provider = CGDataProviderCreateWithURL(url);
document = CGPDFDocumentCreate(provider);
CGDataProviderRelease(provider);
rect = CGRectMake(x, y, w, h);
CGContextDrawPDFDocument(context, rect, document, pageNumber);
CGPDFDocumentRelease(document);
```

PDF Documents



- NEW Import PDF 1.3 documents
- NEW Export PDF 1.3 documents
- NEW Full round-tripping of CG API
- NEW PDF/X-3: 2002

Demo

PDF Support

PDF/X-3 Example

```
auxInfo = CFDictionaryCreateMutable(NULL, 0, …);
CFDictionarySetValue(auxInfo, kCGPDFContextTitle, CFSTR(“My Document”));
…
outputIntent = CFDictionaryCreateMutable(NULL, 0, …);
CFDictionarySetValue(outputIntent, CFSTR(“S”), CFSTR(“GTS_PDFX”));
CFDictionarySetValue(outputIntent, CFSTR(“OutputCondition”), CFSTR(“CGATS
    TR 001 (SWOP)”));
…
CFDictionarySetValue(auxInfo, kCGPDFContextOutputIntent, outputIntent);
context = CGPDFContextCreateWithURL(url, NULL, auxInfo);
/* Don’t forget to release everything that needs releasing…. */
```

Color Spaces

- **CGColorSpace** contains color-interpretation data
- Device-dependent color spaces
- Device-independent color spaces

ColorSync Support

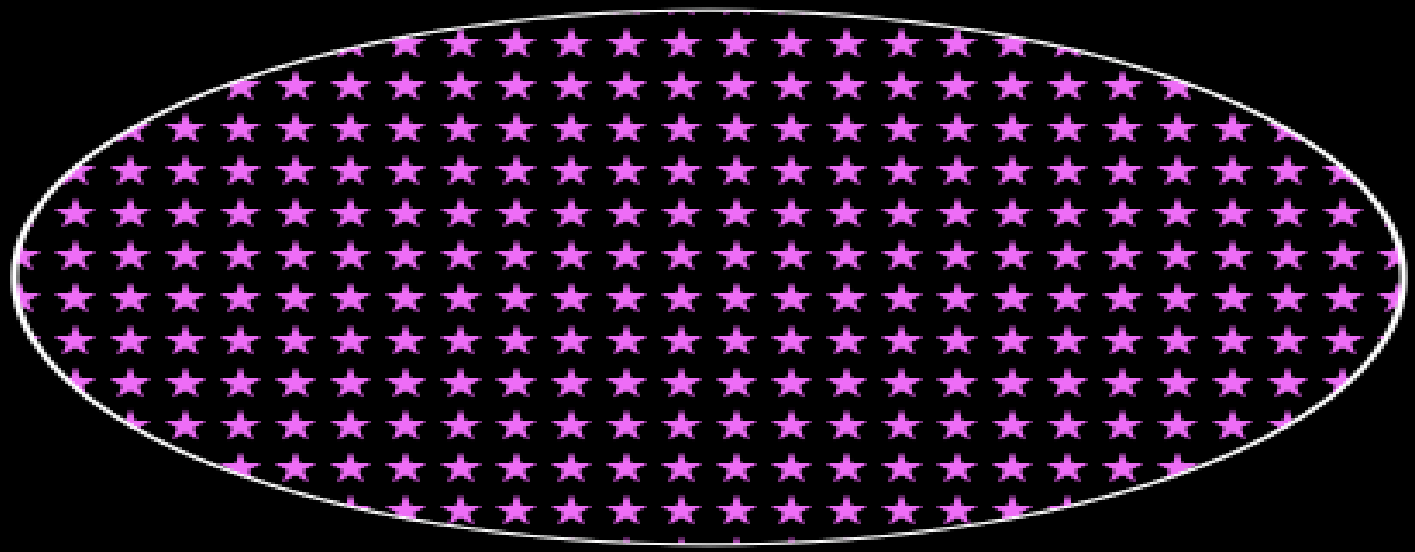
- Additional colorspace convenience functions
  - User's "document default" colorspace
  - Colorspaces for fast on-screen drawing and high-fidelity printing
  - Standard Apple colorspaces
  - API coming soon . . .

Special Color Spaces

- Indexed
    - For example, GIF images
- Pattern

Patterns

- Replicated drawing via callbacks
- Resolution independent
- Better than doing it yourself



Demo

Patterns

Using Patterns

- Create a pattern with CGPatternCreate
- Set the color space to be a pattern colorspace
- Set the pattern in the context
- Fill or stroke a path

Example

```
static const CGPatternCallbacks callbacks = { 0, &drawCircle, NULL};
rect = CGRectMake(0, 0, patternWidth, patternHeight);
pattern = CGPatternCreate(info, rect, CGAffineTransformIdentity,
    xSpace, ySpace, kCGPatternTilingConstantSpacing, false, &callbacks);
baseSpace =    CGColorSpaceCreateDeviceRGB();
colorspace = CGColorSpaceCreatePattern(baseSpace);
CGContextSetFillColorSpace(context, colorspace);
CGContextSetFillPattern(context, pattern, color);
/* (Don’t forget to release colorspaces and pattern….) */
```

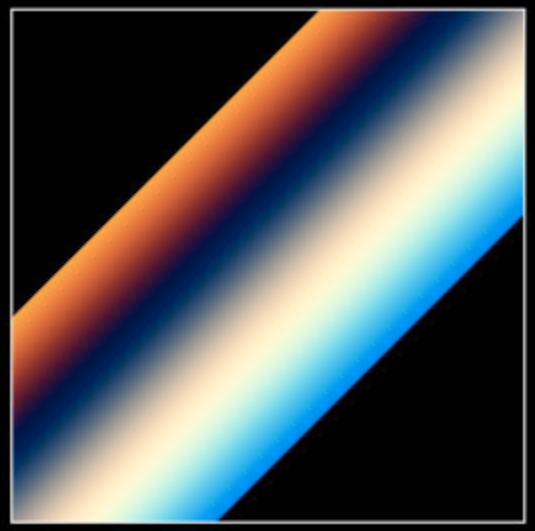
Shadings

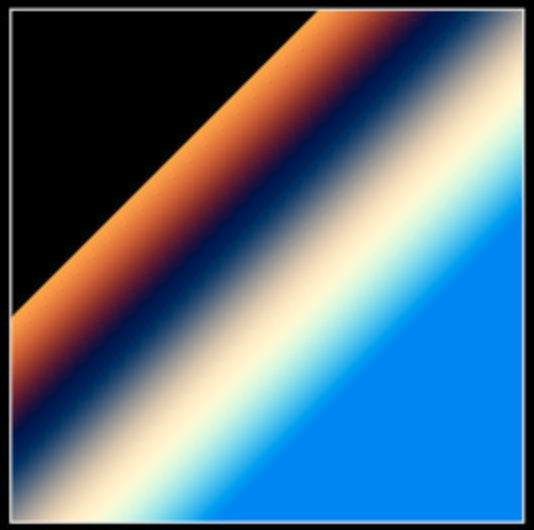
- **CGShadings** represent radial or axial gradients
- Resolution independent
- Better than doing it yourself

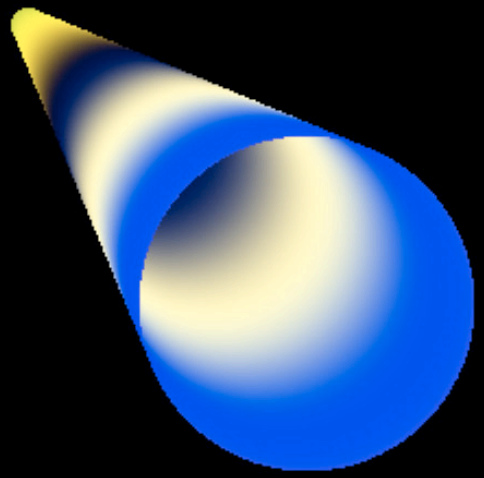
Demo

Shadings

Shading Examples







Shadings

- Not required to be linear
  - Callback determines color along axis or radius
  - **CGFunction**—one value in, color out

Example

```
static const CGFunctionCallbacks callbacks = { 0, &evaluateLinear, NULL };
static const float domain[2] = { 0, 1 };
static const float range[8] = { 0, 1, 0, 1, 0, 1, 0, 1 };
function = CGFunctionCreate(info, 1, domain, 4, range, &callbacks);
```

Using Shadings

- Create a color function with **CGFunctionCreate**
- Create a shading with **CGShadingCreate**
- Draw the shading with **CGContextDrawShading**

Example

```
colorspace = CGColorSpaceCreateDeviceRGB();

shading = CGShadingCreateAxial(colorspace, startPoint, endPoint,
    function, extendStart, extendEnd);

CGContextDrawShading(context, shading);

/* Don’t forget to release everything appropriately…. */
```

Demo

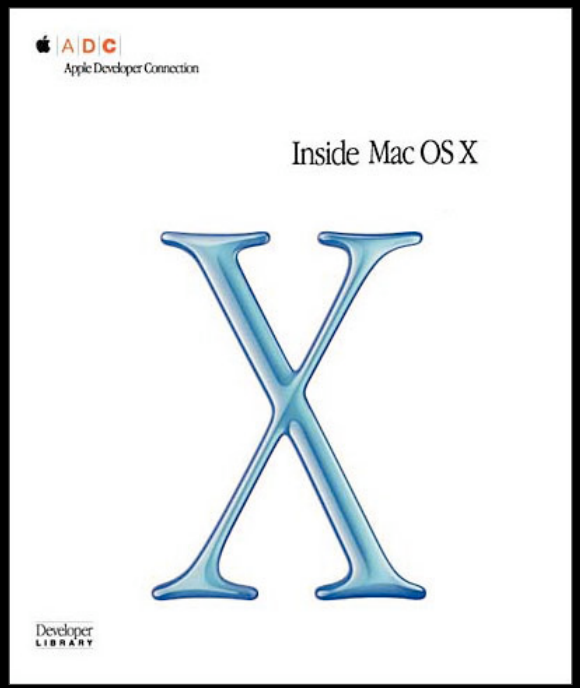
All Together Now…

Summary

- Lots of new API to try
- Continuing to add API where appropriate
- As always, your feedback is encouraged

Documentation

Quartz 2D



- Drawing With Quartz 2D
- Quartz 2D Reference
- Quartz Primer

Documentation > Core Technologies > Graphics > Quartz 2

http://developer.apple.com/techpubs/macosx/CoreTechnologies/graphics/Quartz2D/quartz2d.html

More Documentation

Quartz 2D

- Header file comments (Jaguar APIs)

  /System/Library/Frameworks/ApplicationServices.framework/Frameworks/CoreGraphics.framework/Headers

- Technical Notes: Core Graphics

  developer.apple.com/technotes/indexes/cg-a.html

- Technical Q&A's: Core Graphics

  developer.apple.com/qa/indexes/cg-a.html

Roadmap

500 Graphics and Imaging Overview	Room A2 **Tue., 10:30am**
501 Quartz 2D and PDF	Room A2 **Tue., 2:00pm**
503 Exploring the Quartz Compositor	Hall 2 **Tue., 3:30pm**
504 OpenGL: Graphics Programmability	Room A2 **Tue., 5:00pm**

Roadmap

505 OpenGL: Integrated Graphics I	Room J **Wed., 9:00am**
506 OpenGL: Integrated Graphics II	Room J **Wed., 10:30am**
109 Darwin Printing	Room J **Wed., 2:00pm**
509 ColorSync and Digital Media	Room C **Wed., 5:00pm**

Roadmap

510 Printing and Mac OS X	Hall 2 **Thurs., 10:30am**
513 OpenGL: Advanced 3D	Room J **Thurs., 3:30pm**
514 OpenGL: Performance and Optimization	Room J **Thurs., 5:00pm**
515 Image Capture Framework	Room C **Fri., 2:00pm**

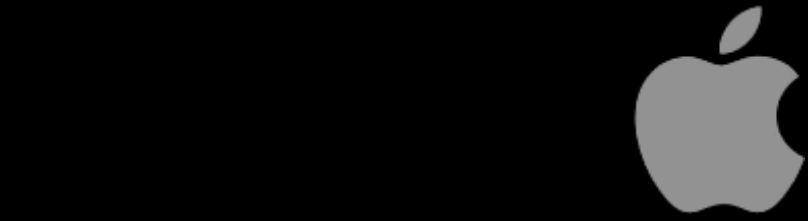
Roadmap

516 Graphics and Imaging Performance Tuning	Hall 2 **Fri., 3:30pm**
Feedback Forum Graphics and Imaging	Room J1 **Fri., 5:00pm**

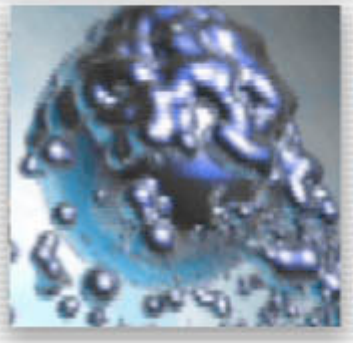
Who to Contact

Travis Brown
Graphics and Imaging Evangelist
travis@apple.com

http://developer.apple.com/wwdc2002/urls.html



Q&A



Travis Brown
Graphics and Imaging Evangelist
Worldwide Developer Relations

http://developer.apple.com/wwdc2002/urls.html


WWDC2002


WWDC2002


WWDC2002